

LASERLINK
GAVIN WILLIAMSON (049) 621678

DISK TOOL KIT
BY BOB KITCH

LASERLINK

20A BRUNKER RD.
BROADMEADOW 2292
PHONE (049) 62 1678

DISK TOOLKIT

Disk Monitor for VZ 200 - 300 Computers

This is an efficient System Monitor to produce and test Z-80 machine code programs / routines and a great aid in program analysis. eg the ROM routines.

The program provides the possibility of transferring BASIC and machine code programs between cassette and disk.

During a program test, breakpoints can be set, which, when reached, transfer control to the monitor. The registers and memory locations can then be inspected, modified if necessary, after which, execution can be continued.

A find routine allows a search for a particular set of data in a selected area of memory.

The Disk Monitor is about 3K bytes long and automatically locates, after loading, to an area at the top of available RAM.

Loading the Program

The monitor will load and run from disk with BRUN"MONITOR". After correct loading, the program's title screen appears. The last symbol displayed, "#", is the prompt, at which you are required to enter a monitor command.

Command Syntax

A command has the form:

x aaaa bbbb cccc (RETURN)

X = command (enter an alpha character)

aaaa bbbb cccc = hexadecimal parameters with a maximum of four digits

The parameters are entered one after another separated by a space. The space is optional between the command and Parameter 1.

After pressing the RETURN key the line entered is analysed and the command executed. If the command isn't recognized the monitor responds with WHAT? #.

Each command can have the parameters omitted. In this case the symbol, ":", must be used. This ensures that the parameters used last, which are temporarily stored in the work area, are utilised. In the absence of the ":", all three parameters will be assumed to be X'0000'.

Each command can be interrupted by simultaneously pressing the CTRL and BREAK keys. Listing can be slowed down with the SHIFT key and interrupted with the ":" key. Continue an interrupted listing with the SPACE key.

Two areas of memory serve the program's internal management. The register preservation area will temporarily store the Z-80 registers, for the program being worked on, after a breakpoint.

An additional work area serves as a temporary store for entered parameters.

The following commands are at your disposal.

G B - Breakpoints can be set
M C - Copy areas of memory
Q D - Disk drive selection
H E - Enter a program with a breakpoint set
P F - Find consecutive bytes
K G - Continue a machine code program
A I - Initialise the monitor work area
J - Jump to a program without breakpoints
N K - Clear an area of memory
F L - Load a program from cassette
D M - Modify an area of memory
C P - Print out an area of memory
S R - Read a program from diskette
E S - Save a program to cassette
B T - List an area of memory
O V - Compare areas of memory
R W - Write a program to diskette
L X - Read / modify the register contents

Monitor Commands

A # I - Initialise the register preservation area and the work area with this command.

With this command the register preservation area and the internal work area can be reset. The I command will be executed automatically at the start of the program. On execution, the message INITIALISED will be displayed.

B # T aaaa bbbb - List the contents of memory to the screen. Displays an area of memory from address aaaa to address bbbb in hexadecimal and in ASCII format on the screen. At the beginning of each line is the memory address followed, in hexadecimal, by 8 data bytes in blocks of four, and then the ASCII equivalent of these bytes. Only hex values from 20-7F will be displayed. For the contents of other bytes the symbol "." will be displayed.

C # P aaaa bbbb - List the contents of memory to the printer. The P command is equivalent in function to the T command, except that the data output is diverted to a connected printer.

D # M aaaa - Modify one or more memory locations. The monitor displays the memory address aaaa and the byte contained in it. Immediately following the # symbol, you can enter the new byte or separate it by one or more spaces. After pressing the RETURN key, the new data is transferred into memory. If the address is located in ROM, or is not found, the message ADDRESS ERROR is displayed. When RETURN is pressed without a new input, the monitor will move to the next memory location. Entering the symbol "." instead of a data byte will cause the monitor to leave the modify mode.

Note: When leaving the modify mode, Parameter 2 is set to the last address being worked on. Using the command "T:", you can immediately view the modified area of memory.

E # S aaaa bbbb cccc name - Save an area of memory to cassette.
With this you can save a machine code program on cassette.

aaaa - Start address
bbbb - End address
cccc - Program type
name - Program name (max. 15 characters)

A program saved in this way can be loaded using the CLOAD or CRUN commands and will autorun.

F # L aaaa+- - Load a program from cassette.
Reads a program from cassette and then displays:

Start address End address Program type Program name

With the optional parameter aaaa, in conjunction with + or -, an offset can be given, to shift a program in memory when loading (+ can fail).

Attention: A loaded program with an offset is not able to be run, unless there is an adjustment to jumps, otherwise an address error will result.

G # B aaaa - Define a breakpoint.
With this command, a breakpoint can be defined to test a program. This is set in a program with the command E or G. On reaching this address the tested program will jump to the monitor, and the program registers will be preserved. This is announced with "BREAK IN aaaa". Memory and register contents can be inspected and modified and afterwards the program can be resumed.

H # E aaaa - Execute a machine code program from the address aaaa. A breakpoint defined with the B command will be set in the program. The program registers will be loaded with the data from the register preservation area.

I # J aaaa - Jump to a machine code program.
Unlike E, however, no breakpoint will be set.

K # G - Continue a machine code program.
This command will enable a program to be resumed after a breakpoint interruption. If necessary, define a new breakpoint, as long as it is not identical to the continuation address.

Note: On jumping to the routine being tested with the monitor, the stack pointer value will be set to the start address of the monitor. This however, can be altered from the routine being tested, at any time.

- L # X - Display and alter registers.
 With this the register preservation area can be directly accessed, whilst the program registers are temporarily stored after a breakpoint interruption.
 Entering # X/XX causes the contents of the register pair (XX) to be displayed and following the symbol #, the new data can be entered (similar to the M command).
 The command # X: prints the entire register preservation area. The flag register will be interpreted and the set flags displayed in abbreviated form.
- M # C aaaa bbbb cccc - Copy an area of memory.
 Copy an area of memory from aaaa with a length of cccc to bbbb. Both areas should not overlap.
- N # K aaaa bbbb cc - Clear memory.
 All memory locations from aaaa to bbbb are overwritten with the value cc.
- O # V aaaa bbbb cccc - Compare memory.
 Compare two areas of memory at aaaa and bbbb to a length of cccc. If there is a difference, then the differing contents and their associated addresses will be displayed.
- P # F aa bb cc dd ee ff .. nn - Find a series of bytes.
 Search memory for consecutive bytes aa bb cc .. nn. The search begins at address 0000. If the data is found, the monitor changes to modify mode (see the M command) and displays from the start address of the data. Otherwise, the message NOT FOUND # will result.
 If the character ":" is entered after the command symbol, then the search is executed from the memory address in Parameter 1. As each find occurs, Parameter 1 is set to that address location to enable the search of the memory to continue.
- # aaaa bbbb cccc - Set parameters command.
 With this command the parameter area can be set. This, for example, is of use if the search with the F command is to begin from a particular address.
- Q # D n - Disk drive selection.
 With the D command, you can, during execution of the monitor, switch between the two connected disk drives. Ensuing "R" and "W" commands will execute to the drive selected.
 n = Drive number (1 or 2)

R # W aaaa bbbb cccc name - Write an area of memory to diskette.

With this an area of memory (BASIC or a machine code program) can be saved to disk.

aaaa = Start address

bbbb = End address

cccc = Program type

name = Program name (max. 8 characters)

Valid program types:

00F0 = BASIC program

00F1 = Machine code program

In this way a saved program can be loaded or run following the normal load or execute commands of the DOS (LOAD, RUN, BLOAD, BRUN).

Possible error messages:

INPUT OUTPUT ERROR

During reading or writing of a sector an error is encountered.

FILE ALREADY EXISTS

The filename entered already exists on the disk.

DIRECTORY IS FULL

No room is found in the disk directory to contain the new file.

DISKETTE FULL

There aren't enough free sectors available on the disk to save the program.

15 # R name - Reads a program or data block from disk and displays the following:

Start address End address Program type Program name

The program name has a maximum of 8 characters.

Possible error messages:

READ ERROR

An error (address mark not found or checksum error) occurs when reading a sector from disk.

INCORRECT FILETYPE

The file with the name entered is actually found on the disk, however it cannot be loaded, because of the file type (Type D).

FILE NOT FOUND

The file with the name entered is not found on the disk.

DISK DOCTOR

With this program, you can physically work on your disks for the VZ 200 and VZ 300 computers.

You can directly read each sector on the disk, have it displayed in hexadecimal, make modifications and write it back to disk.

Allows any checksum errors on the disk to be amended, as long as there is no physical damage to the magnetic layer, by reading the defective sector and then writing it back correctly.

Operating directions

The program is loaded from disk and run with BRUN"DOC". After correct loading, a title screen appears, which requests from you the address details necessary to read the first sector.

Drive (1/2)

Select the desired drive by entering "1" or "2".

Track (00-27)

Enter the desired track on the disk in hexadecimal notation. Two digits must always be entered eg. 0F and not just F.

Sector (0-F)

Enter in hexadecimal the sector to be addressed on the track selected above.

Each time the permissible value range is given in brackets. All other input except this, CTRL-BREAK and "Q" will be ignored.

Entering "Q", instead of the input outlined above, will end the program and reinitialise the computer.

Simultaneously pressing of "CTRL" and "BREAK" at any point will result in a restart, and the title screen will again appear. You must, for example, use this method if you wish to change to the other disk drive.

Correctly enter all address values for a sector and it will be read from the disk and displayed on the screen.

This display is made up as follows:

In the left three columns, the track and sector numbers are given in hexadecimal.

On the right next to them are the relative sector addresses, two digit hexadecimal numbers from 00 to 7B. These always refer to the first byte of each line. Each eight bytes of the sector is listed in hexadecimal notation, in two four byte blocks per line.

In the right hand eight columns there follows the ASCII equivalent of these eight bytes. Those ASCII codes not able to be displayed are replaced by ".".

+ or RETURN - Move forward ie. the next sector on the disk will be read and displayed.

- - Move backwards ie. the previous sector on the disk will be read.

A - New track and sector address. You will be asked to enter the TRACK and SECTOR again.

M - Make changes in the MODIFY mode. You can alter the sector contents and then write the sector back to disk.

CTRL BREAK - Restart the program.

When reading a sector you can encounter the following errors.

SECTOR NOT ADDRESSED

Address data marks of the addressed sector cannot be found on the disk.

After pressing the RETURN key the program returns to the track and sector selection.

CHECKSUM ERROR ENCOUNTERED

During reading of the sector a checksum error is detected in the data field of the sector. After pressing RETURN the sector contents will be displayed and can then be worked on. When this sector is written back the checksum will be automatically corrected.

Operating directions in MODIFY mode.

When you have the sector display, press the M key to change the program to the MODIFY mode.

It is recognizable, in that one of the displayed digits in the sector contents flashes.

In this mode you can alter the sector contents and write the sector back to disk.

The following operating functions are at your disposal.

Arrow keys (+CTRL) - With these you can position the flashing cursor to the desired place in the data field of the sector.

O-F - Pressing one of these keys alters the sector contents in the internal buffer. The ASCII display on the right side is automatically altered with it.

Q - Pressing Q returns you to the display mode, all alterations to the sector will be cancelled.

CTRL BREAK - Restarts the program.

RETURN - Write the sector back to disk.

You will be asked next if all alterations are correct. The "Y" key will write the sector to disk. Pressing the "N" will return you to the MODIFY mode. After a sector is written to disk, it will, on completion, be shown on the screen in the display mode.

By entering a new address or by stepping forwards or backwards, another sector can be displayed.

When writing a sector, just as when reading, an error can occur, where the address of a sector isn't found. In such cases, an error message is displayed. After pressing RETURN another attempt will be made to read the sector.

COPY DISK

A disk copy program for the VZ 200 and VZ 300.

COPY DISK enables quick and sure copying of disks for the VZ 200 and VZ 300 computers with one or two connected disk drives.

Before the copying process an initialisation of the destination diskette is carried out.

COPY DISK contains, for initialising disks, in contrast to the standard initialisation, an improved formatting, which results in a considerable time saving (about a 1/3), during writing or reading of data and programs.

A further quickening of the copying process is achieved, in comparison to DCOPY, in that only the used tracks on a disk are copied.

All copied tracks will be verified afterwards.

COPY DISK furthermore, through detailed user directions, makes possible problem free centreing of the diskette, when it is inserted in the drive.

Operating directions

COPY DISK can be loaded from disk and run with BRUN"COPY".

After loading, a title screen appears, which asks, first of all, for the drive number prior to copying.

The given value can be selected with the <RETURN> key, or overwritten by pressing the "1" or "2" keys.

If there is only one drive, enter the same digit in both cases.

At this point, by pressing the "Q" key, the program will end and the computer will be reinitialised.

Pressing the <CTRL> <BREAK> keys now, or at any time, causes a break in the current function and a restart of the program.

After drive selection, a request is made to insert the diskette.

This is carried out in two steps, to ensure centreing on the drive cone.

1. Insert the diskette without closing the drive door (locking lever).

After pressing <RETURN> the drive starts up.

2. Close the drive door after the motor is running. In this way, the diskette is drawn in a controlled way exactly onto the cone.

After pressing the <RETURN> key once more, the program continues.

When using two disk drives, the diskette need only be inserted once at the beginning of the copying process.

When using only one disk drive the diskette is changed many times. At each change, carry out the steps outlined above.

The actual copying process is carried out in four steps.

1. Initialise the destination disk.
2. Read the data from the source disk.
3. Write the data to the destination disk.
4. Verify the data on the destination disk.

Steps 2-4 are repeated several times, if necessary. This is dependent on the size of the available memory and the number of used tracks on the source disk.

At each step, the relevant information is displayed and the track number indicated.

On completion of the copying process, you can either make a further copy of the source disk (enter "Y") or restart the program (enter "n").

During the copying process, if a write or read error is encountered on either of the two diskettes, then the relevant error message will be displayed. Pressing the <RETURN> key, will return you to the start of the program.

As mentioned already, the initialisation of the destination diskette carried out by COPY DISK, makes possible considerably faster data access.

You can use COPYDISK's improved initialisation on your other diskettes too.

Enter the same drive number for the source and destination disks. You will first be asked to insert the destination diskette, which, after insertion, will be immediately initialised.

On completion of the initialisation, you will be invited to insert the source disk. Responding with <CTRL> <BREAK> will return you to the start of the program.

*data files loaded 1.0 P
space between parts of file is 20 20 00 20.
to merge all the P in file - 1.0 delete 20 + P# between files
by inserting 20. Required P*